IN THE CLAIMS:

Please amend the claims as follows:

1. (Twice Amended) A program converting unit for generating a machine language instruction from a source program for [an embedded microprocessor series that manages an N-bit address while processing M-bit data,] a microprocessor having an address width N and a data width M, N being greater than M, N and M being [customized] inputs to the program converting unit as specified by a user, the value of N depending on a program size of the source program, said program converting unit comprising:

parameter holding means for holding a data width M and a pointer width N, said data width M representing the number of bits of data used in the source program, said pointer width N representing the number of bits of an address, said N and M being [designated] input by a user during an execution of the program converting unit, the value of N depending on the program size; and

generating means for generating an instruction to manage said data width M when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width N when a variable operated by said instruction represents the address,

wherein the program converting unit generates a unique set of machine language instructions from the source program for each N specified by the user.

6. (Amended) A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediary instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address;

parameter holding means for holding a data width and a pointer width, said pointer width designated by a user as an input during an execution of the program converting unit, said data width representing the number of bits of the data while said pointer width [representing] represents the number of bits of the address; and

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address.

---

13. (Twice Amended) A program converting unit for generating a machine language instruction from a source program for an embedded microprocessor series that manages an N-bit address while processing M-bit data, N being greater than M, N being [customized] an input to the program converting unit depending on a program size, said program converting unit comprising:

parameter holding means for holding a data width M and a pointer width N, said data width M representing the number of bits of data used in the source program, said pointer width N representing the number of bits of an address, said N being [designated] input by a user during an execution of the program converting unit, the value of N depending on the program size;

generating means for generating an instruction to manage said data width M when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width N when a variable operated by said instruction represents the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation; and

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate, wherein the program converting unit converts the source program into one of a plurality of different machine language programs depending on the values of N and M.

---

20. (Twice Amended) A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediate instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address, and one of signed and unsigned data;

parameter holding means for holding a data width and a pointer width designated by a user during an execution of the program converting unit, said data width representing the number of bits of the data, said pointer width representing the number of bits of the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address; and

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate.

---

27. (Twice Amended) A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediary instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address, and one of signed and unsigned data;

parameter holding means for holding a data width and a pointer width designated by a user during an execution of the program converting unit, said data width representing the number of bits of the data, said pointer width representing the number of bits of the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address;

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate, wherein said generating means includes:

determining means for determining a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to a memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective memory-access width depending on the type of a variable to be accessed shown in said table;

register managing means for outputting a direction, in case of the (2) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective bit-width depending on the type of a variable to be read/written from/in the register shown in said table;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage a corresponding bit-width held in said parameter holding means for the immediate as an effective bit-width depending on the type of the immediate shown in said table; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means, and wherein

said compensate instruction generating means includes:

instruction judging means for judging an arithmetic operation instruction that will possibly cause an overflow for all the machine language instructions when said option instructing means holds the user's direction for executing the overflow compensation;

determining [judging] means, with respect to a variable in the arithmetic operation instructions determined by said instruction determining means, for determining an effective bit-width and whether said variable is signed or unsigned by referring to said table;

sign-extension instruction generating means for generating a compensation instruction in case of a signed variable, a logical value of a sign bit being filled into all bits higher than the effective bit-width in a register that is to store said signed variable by said sign-extension compensation instruction; and

zero-extension instruction generating means for generating a zero-extension compensation instruction in case of an unsigned variable, a logical value "0" being filled into all bits higher than the effective bit width in a register that is to store said unsigned variable by said zero-extension compensation instruction.

28. Twice Amended) A processor being one out of an embedded custom processor series of processors with different address bit widths, having an address bit width which can be [customized] selected by a user in accordance with a program size, comprising:

memory means for storing a program, the memory means having a minimum storage capacity of 2^N bytes to store the program and having N addresss lines, the program including an N-bit data arithmetic operation instruction and other instructions operating on both N-bit and M-bit data, N being greater than M; and

a processor core having an address bus of N bits which is equal in size to the number of address lines of the memory means, the processor core being selected from a plurality of processor cores and comprising:

a program counter for holding an N-bit instruction address to output [the same] an instruction at the N-bit address to the [said] memory means, the program counter having an N-bit address length which is equal in size to the number of address lines of the memory means;

fetching means for fetching an instruction from [said] the memory meansusing the N-bit instruction address from said program counter; and

executing means for executing all arithmetic operation instructions at N-bit length and for executing other instructions [aside from] except for arithmetic operation instructions at one of N-bit length and M-bit length, the executing means having N-bit length;

[wherein said memory means has a storage capacity equivalent of up to 2^N bytes, and] whereby an N-bit address is calculated by the N-bit arithmetic operation independently of a data bit-width, said data bit-width being M.

35

~~44.~~ (Twice Amended) A processor for operating certain data in accordance with an instruction in a program, comprising:

a first register means for holding N-bit data;

a second register means for holding N-bit data;

sign-extending means for extending said M-bit data to N bits by copying an MSB of said M-bit data in a direction of an upper order, M being less than N;

zero-extending means for extending said M-bit data to N bits by copying a value "0" in a direction of an upper order;

operating means for operating an arithmetic operation in accordance with an instruction;

instruction control means for decoding an instruction to zero-extend M-bit immediate data when said M-bit immediate data are to be stored in said first register means by the decoded instruction and to sign-extend said M-bit immediate data when said M-bit immediate data are to

be stored in said second register means by the decoded instruction, said zero-extended and sign-extended N-bit immediate data being outputted in one of two methods, one method being to send the extended N-bit immediate data from their respective extending means to their respective register means directly, the other being to send the same via the operating means to their respective register means, with said instruction including an indication for storing in the first register means and said instruction including an indication for storing in the second register means being of two different kinds of [instruction] instructions, both kinds of instructions having a same operation code but having different destination operands.

Claim 47 (Twice Amended), line 23, delete "both".

Claim 54 (Twice Amended), line 6, delete "customized" and insert --input--.